


# The Design of a Hybrid Model-Based Journal Recommendation System

ADEWALE OGUNDE

## Related papers

[Download a PDF Pack](#) of the best related papers 



[Realizing an efficient IoMT-assisted Patient Diet Recommendation System through Machine...](#)

Fazal Noor

[Building Recommender Systems for Network Intrusion Detection Using Intelligent Decision Technolo...](#)

Mrutyunjaya Panda

[A Review of Recommender Systems for Choosing Elective Courses](#)

Mfowabo Maphosa

## The Design of a Hybrid Model-Based Journal Recommendation System

Adewale Opeoluwa Ogunde\*, Mba Obasi Odim, Oluwabunmi Omobolanle Olaniyan, Theresa Omolayo Ojewumi, Abosede Oyenike Oguntunde, Michael Adebisi Fayemiwo, Toluwase Ayobami Olowookere, Temitope Hannah Bolanle

Department of Computer Science, Redeemer's University, Nigeria

### ARTICLE INFO

#### Article history:

Received: 29 September, 2020

Accepted: 16 November, 2020

Online: 14 December, 2020

#### Keywords:

Academics

Collaborative filtering

Content-based filtering

Journal

Publications

Publication Authors

Recommender systems

### ABSTRACT

*There is currently an overload of information on the internet, and this makes information search a challenging task. Researchers spend a lot of man-hour searching for journals related to their areas of research interest that can publish their research output on time. In, this study, a recommender system that can assist researchers access relevant journals that can publish their research output on time based on their preferences is developed. This system uses the information provided by researchers and previous authors' research publications to recommend journals with similar preferences. Data were collected from 867 respondents through an online questionnaire and from existing publication sources and databases on the web. The scope of the research was narrowed down to computer science-related journals. A hybrid model-based recommendation approach that combined Content-Based and Collaborative filtering was employed for the study. The Naive Bayes and Random Forest algorithms were used to model the recommender. WEKA, a machine learning tool, was used to implement the system. The result of the study showed that the Naïve Bayes produced a shorter training time (0.01s) and testing time (0.02s) than the Random forest training time (0.41) and testing time (0.09). On the other hand, the classification accuracy of the Random forest algorithm outperformed the naïve Bayes with % correctly classified instance of 89.73 and 72.66; kappa of 0.893 and 0.714; True Positive of 0.897 and 0.727 and ROC area of 0.998 and 0.977, respectively, among other metrics. The model derived in this work was used as a knowledge-base for the development of a web-based application, named "Journal Recommender" which allowed academic authors to input their preferences and obtain prompt journal recommendations. The developed system would help researchers to efficiently choose suitable journals to help their publication quest.*

## 1. Introduction

The overload of information available through the world wide web today has created a lot of challenges for information seekers. Researchers who are in need of journals related to their research area sometimes find it difficult to access and keep track of the relevant journals or research materials of their interest including searching for relevant journal to publish research outputs.

Conducting a very good research could be very stressful and searching for the right journal to publish it could even be more stressful. Academic authors are therefore faced with making their decision based on a number of competing preferences such as the

scope of the journal, the access type, article processing charge, speed of the review process, impact factor, indexing and ranking just to mention a few. A simple approach used in searching for such journal publishers is to randomly surf the web by typing some specific keywords predetermined by the author. Results obtained from using this style may not be too beneficial as it is probability-based and highly dependent on the searching skills and maneuvering abilities of the author. Some authors, also, make their journal selection decision by checking out the reference lists form existing related journals available to them. This style is also limited in scope and it depends fully on related journals currently available to the author. Moreover, some potentially useful journals may be difficult to access by authors. It is also a fact that the article processing charge of some accessible journals may be too expensive especially for authors from low income countries where

\*Corresponding Author: Adewale Opeoluwa Ogunde, Department of Computer Science, Redeemer's University, Ede, Osun State, Nigeria, +2347036090090, ogundea@run.edu.ng

access to research grants and funding are also a major challenge. There is, therefore, a need for a recommender system that can assist in solving the highlighted problems.

Recommender systems are described as information filtering systems that are designed to solve the challenge of information overload by filtering a large amount of information according to the user's observed behaviour, interest, preferences and producing very useful and vital information fragment as output [1]. Recommender systems have the ability to predict the choice of its user based on their past preferences. In this work, the use of a journal recommender systems to automatically suggest relevant journals to the researchers based on some initial information provided by the users and other information from journal publishing sites, was developed. To provide more relevant recommendations, the developed recommender system would incorporate the list of journals, the type of journal access whether it is an open access or restricted access, free or article processing fee, research topic, the field of study, indexing and rating. It is also specific to the field of study, which is Computer science-related journals.

This work resolves information overload problem to academic authors; whereby a researcher would want his/her authored works to be published, this might take a more extended period because the scope in the popular bibliographic database is vast. Also, in this age, when there is high proliferation of academic journals, getting the right journal to publish an article could turn out to be a huge task. A recommender system for guiding researchers when searching for journals to publish their articles promptly was developed in this work. This would make their publication quest easier and effective.

## **2. Literature review**

In this section, some relevant literature tied to the research were reviewed and some closely related works were also presented.

### *2.1. Recommender Systems*

The increasing use of internet services has resulted in a massive amount of data and information available on the internet. The nature of this data is so dynamic as well as diverse that it becomes difficult for one to look for the kind of data that matches the requirements. This challenge has aroused the development of new technology to assist Internet users to cope with information overload and indecisions on a product, service and items. They are called Recommender Systems. According to [2], recommender systems are applications that filters users' preferences, suggesting the most suitable items to specific users based on similar information about the users, the items, and their relationships. Solving the problem of information overload and filtering out the most important items, products and services are considered to be the primary purpose of recommender systems. Personalized recommendations could be generated by analyzing the behaviour of a user and other similar users to suggest user's interests and preferences [3]. Recommender systems are very useful in diverse domains such as education, commercial, entertainment, for example, movies, music and others. This work focused on deploying the recommender system in the academic domain in order to facilitate the search for useful educational web resources such as Journals, more comfortable to find, thereby helping

researchers enhance their publishing process by getting access to very useful journal recommendations leading to faster publication time.

### *2.2. Techniques of Recommender Systems*

The main recommendation techniques include traditional methods such as content-based, collaborative filtering-based, hybrid-based (a combination of content-based and collaborative filtering methods) and knowledge-based methods [4]. Other recently appearing variants in the literature are trust-based, group-based, fuzzy set-based, context awareness-based and social network-based methods. The most acceptable and widely used out of all are the traditional recommendation methods. Content-based is further divided into tag-based and descriptive-based. Collaborative filtering's subdivisions are Model-based and Memory-based, under model-based are Clustering, Association, Bayesian networks while for Memory-based are User-based and Item-based.

In content-based recommendation methods, items similar to items previously preferred by a specific user are usually recommended [5]. More on the basic principles, techniques, strengths and limitations of content-based recommender approaches can be found in [5]-[7]. On the other hand, collaborative filtering is a process of filtering information or pattern based on the opinion of other users, or the similarity between items [6]. This approach makes recommendation by finding similar users having a rating history similar to the new user. This implies that user preferences, in this case, are compared with the other users [7]. More information on the collaborative approach, its characteristics, techniques, strengths and drawbacks can be found in [6]-[10]. A hybrid recommendation system is an approach that deals with the combination of different recommendation models to evolve a more efficient and effective technique that would outperform the traditional stand-alone recommendation techniques. This is done to overcome the weaknesses of stand-alone recommendation techniques while combining their power and strengths to achieve better recommendation. There are seven main combination approaches for building hybrid recommendation system, and they are also grouped into three designs which are Monolithic, Parallelized and Pipelined. More details on hybrid recommendation systems, adapted in this work, could be obtained from the works of [4], [10] and [11].

### *2.3. Related Works*

Apart from some of the previous works done by authors [12] and [13] on the exploration of recommender systems in the education sector, some other related works are also presented here. The authors considered only one algorithm in their implementations. A knowledge-base of product semantics recommender system was built by [14]. The recommender system was capable of determining semantic attributes of products, thereby helping the authors to understand the customer's taste. In order to learn the attributes of the product descriptions found on retailers' websites, supervised and semi-supervised learning techniques were applied to the system. The system was able to map products to an abstract layer of semantic features by recommending other items in the same class of products that

matches the user's model and understand the customer taste and also recommend items across categories. This approach is believed to enhance the user experience and build the user's confidences in the recommendations. The work of [14] focused on narrow classes of products from retailers' websites. A recommendation system based on collaborative filtering approach using Convolution Deep Learning Model was proposed by [15]. The convolution Deep learning model used was based on the Label Weight Nearest Neighbor with three categorizations. The authors in [15] concluded that their proposed system outperformed the other traditional recommendation algorithms in terms of efficiency and effectiveness.

In [16], a product recommender system based on a hybrid recommendation technique was presented. The system focused on books as the product, and in order to recommend more accurately, a personalized book recommendation algorithm was used. It was based on the time-sequential collaborative filtering and content-based filtering. Essential factors such as time sequence of purchasing books of different users in the database and content of the user profile were considered in the system, and it is expected to satisfy users by providing best and efficient book recommendations. The accuracy of recommendation results in this work was low and this was due to the small volume of books used as the experimental dataset. The authors in [17] were able to predict fake information and fake Facebook accounts using a machine learning-based recommendation approach, which can also work on online social network. The limitation of the work was the low prediction accuracies recorded. In [18], a collaborative filtering job recommender system was developed, which was executed by exploring and analyzing Google employees' features to match the qualifications of job seekers and, thereafter, recommending Google to job seekers. The classification algorithms used were SVM, Naïve Bayes, Neural Networks and Decision Tree. The low F1 score and the few number of companies with few features was the limitation to the work.

### **3. Methodology**

This section presents the research methodology, which includes the techniques used in the recommendation system and the design of the hybrid recommendation system.

#### *3.1. Data collection and description*

The primary aim of this work is to develop a system that recommends computer-related journals to academic researchers and authors. Two sets of data were used for the study. First primary data was collected through an online questionnaire prepared through google form and broadcasted across the web. Secondary data was also collected directly from major journal databases and indexing organizations available online. The sources of the data were obtained from bibliographic databases, journal publishers, and their journals, some of them are, Science Direct, Scopus, Researchgate, Elsevier, Google Scholar, DOAJ just to mention a few. The selected journals from these databases were specific to the field of computer science, and the topics selected covered every aspect of computer science including Computer programming, Big data analytics, Software engineering, Computer Architecture, Artificial Intelligence, Cybersecurity, Decision trees, Encryption, Data Mining, Recommendation systems, Human-Computer Interaction, Robotics and Cloud computing just to mention a few.

The questionnaire, used to collect the data was administered online using Google Forms, with eight hundred and sixty-seven (867) respondents, who are researchers that have published at least one article in a standard journal responded to the survey. There were three major sections in the online questionnaire. The first section contained the "Demographic data of the respondents". The second section contained "Questions on Articles Already Published by Respondents in Learned Journals" and "Questions on Publishing Preferences". The first section consisted of five questions. The second and third questions consisted of ten questions, each making a total twenty-five questions altogether. The demographic section contained five questions that required answers about the target audience which included their optional full names, institution, gender, highest academic qualification and work status of respondents. In summary, the majority (71%) of the respondents were males, and the remaining 29% were females. 76% of the respondents were PhD holders, and most (73%) of the respondents had published at least two articles in learned journals.

#### *3.2. Attribute Representation and Categorization*

Four major attributes were used for the recommendation in this work. They are Area of Research Interest (ARI), Journal Access Type (JAT), Article Processing Charge (APC), Indexing and Ranking (IAR). All the data were stored in Microsoft Excel and saved as a comma-separated (.CSV) files. The attributes used were assigned labels represented by three-character nominal values to make the data suitable for classification. These attributes represent preferences of future authors that will be matched in the journal recommendations model built from preferences selected by previous authors and secondary data obtained from publishing sites.

Eighty-one ARI was identified in Computer Science. They are Artificial Intelligence, Adaptive computing, Big Data Analytics, Bioinformatics, Biometric Technology, Classification, Cloud Computing, Clustering, Computational Biology, Computer Forensics, Computer Security, Computer Vision, Content Management Systems, Crowdsourcing, Cryptography, Cybernetics, Data Security, Data Management, Data Warehousing, Data Mining, Decision Support Systems, Decision trees and forests, Distributed systems, E-Commerce, E-Government, E-Learning Technologies, Embedded Systems, Evolutionary computing, Genetic Algorithms, Global Positioning System (GPS), Global System for Mobiles (GSM), Geographical Information Systems (GIS), Green Computing, Grid Computing, Human-Computer Interaction (HCI), Image processing technologies, Information systems, Internet Telephony, Intrusion Detection Systems, Knowledge based systems, Knowledge Management Systems, Machine Learning, Management Information Systems, Mobile agents, Mobile Computing, Multi-agent systems, Network Security, Artificial Neural Networks, Open Source Technology, Parallel Computing, Computer Architecture, Parasitic Computing, Pattern Recognition Techniques, Pervasive Computing, Proactive Computing, Real-time information systems, Recommender Systems, Robotics, Social Networks and Online Communities, Soft Computing, Support Vector Machines (SVM), Theoretical Computer Science, Internet of Things, Virtual Reality, Visualization, Voice Over Internet Protocol, Web-mining, Network Topology, Biosystems & Computational Biology, Cyber-Physical Systems and Design

Automation, Computer Engineering, Database Systems, Computer Education, Computer Graphics, Process Mining, Operating Systems, Programming Systems, Scientific Computing, Security, Theoretical Computing, Networking and Deep Learning. The JAT was divided into two categories, which are Open Access (OA) and Restricted Access (RA). Table 1 shows the distribution characteristics of the preferences of the respondents concerning JAT. The majority (62.51%) of the respondents preferred to publish in open access journals.

Table 1: Journal access type preferences

JAT	Frequency	Percentage (%)
OA	542	62.51
RA	325	37.49
TOTAL	867	100.00

APC was sub-divided into three other categories, which are low, medium and high with low having highest percentage (83.04%) preferences among the respondents, as shown in Table 2.

Table 2: Journal Article Processing Charge Preferences

Category	APC	Frequency	Percentage (%)
BELOW 500USD	LOW	720	83.04
500 - 1000 USD	MEDIUM	92	10.61
ABOVE 1000 USD	HIGH	55	6.34
	TOTAL	867	100.00

Furthermore, few (23.07%) of the respondents did not care about indexing and ranking of the journal while the majority (76.93%) considered ranking and indexing as the most crucial factor for publishing their articles in a journal (Table 3).

Table 3: Journal Indexing and Ranking Preferences

IAR	Frequency	Percentage (%)
YES	667	76.93
NO	200	23.07
TOTAL	867	100.00

3.3. The Design of the Recommender System

This section gives an accurate description of how the proposed system was designed. Unified Modeling Language (UML) was used to understand the flow of this system, and it shows the functions the users can do. The users of the system are represented as actors of the system, and the actions performed are called the use-case. There are two primary users of the system, the admin and the user. Figure 1 shows the use-case diagram for the Admin and User of the system. The Admin will normally log in as the system administrator. The System Admin can add journals and make it

accessible to the system users with all the required details attached to it. The details are its Title, Topic category, Publisher, Its type of access, Its article processing fee, Issues per year and its link for easy access. System Admin can also add the category of a journal which also means the field of the journal, Publishers of Journals into the system and can View, Edit, Update and Delete Journals details. The User, on the other hand, can register and login to the system, search for Journals, Publishers, Categories with the use of keywords, view system recommendations regarding journals selected and generated by the system based on particular preferences.

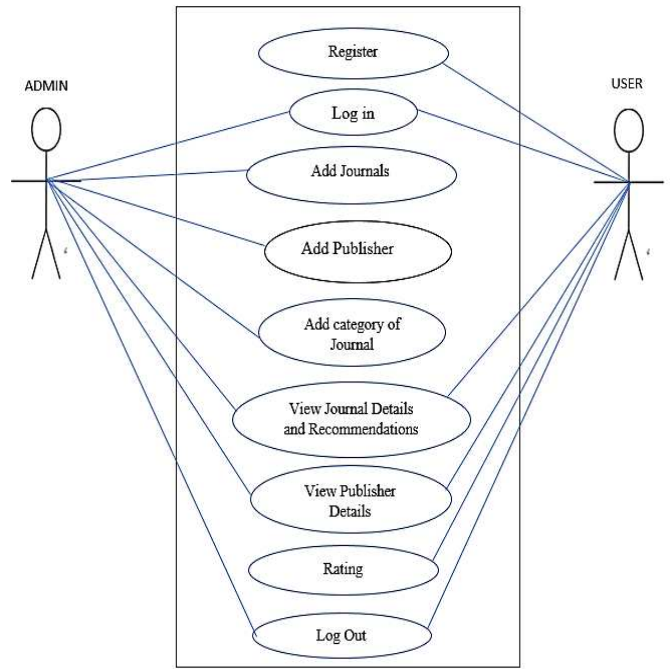


Figure 1: The Use-Case Diagram for the Recommendation System

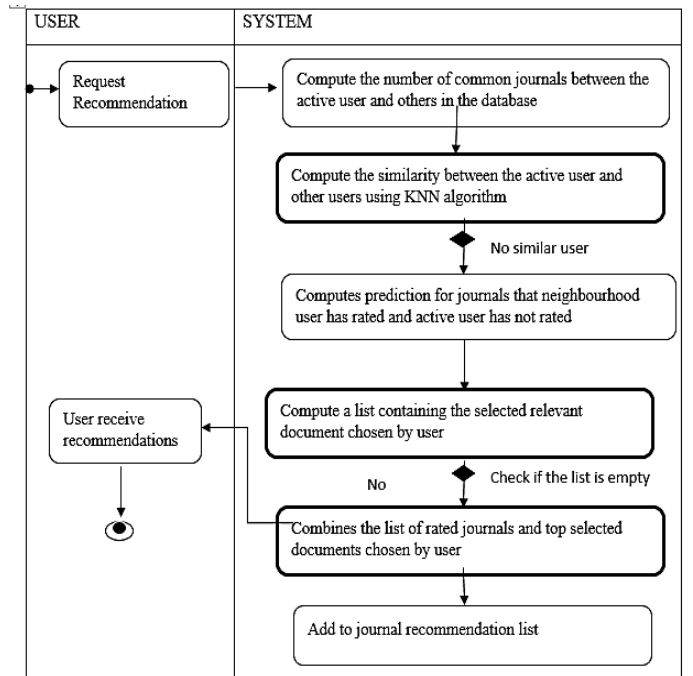


Figure 2: The Hybrid Filtering Recommendation Activity Diagram

3.3.1. Description of the hybrid filtering recommendation activity diagram

In a hybrid recommendation model shown in figure 2, recommendations were based on content-based filtering and collaborative recommendation combined using a monolithic feature augmentation approach. It also employed the combination of Naïve Bayes and Random forest algorithms for searching the desired journal. This model combined the results of collaborative and the result of content-based views the results to the users.

3.3.2. Description of the register process activity diagram

The activity diagram in Figure 3 shows that for a user to make use of the system, the user has to be registered with valid registration details. The user views the blank registration form and fills in personal details such as name, email address and phone number and other details the system requires. As the user begins to fill the form, the system simultaneously stores the information. The user is also expected to fill the preferences details such as the user's research topic, the type of access to journals, and field of study. Then the account form as part of the registration is the authentication using passwords. After the user has filled all the dialogue boxes for registration, the user's account is successfully created, and the account is active.

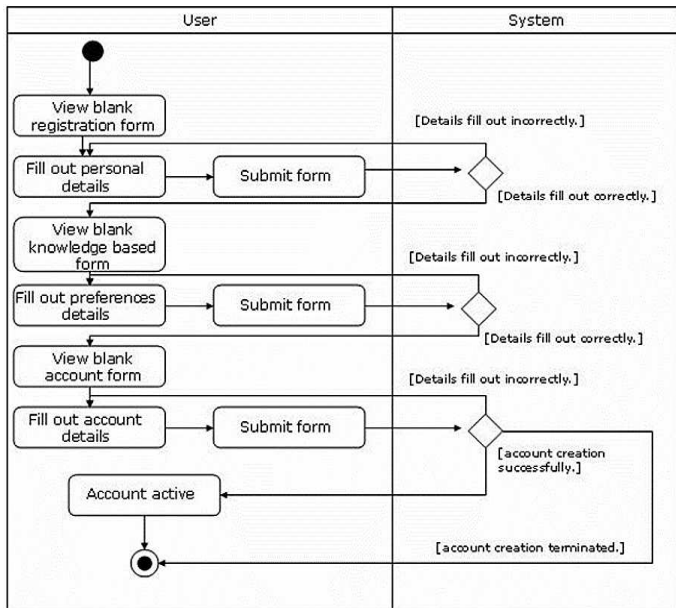


Figure 3: The Register Process Activity Diagram

3.3.3. Description of the search process activity diagram

The activity diagram in figure 4, shows the search process of when a user decides to search for a journal by using its name, publisher and the category, type of access, issue per year. A user can also make use of keywords as an alternative for the searching process, for example, searching for the word " Inter" for "International". The user selects the search form and fills it, the system checks for the search result, filters, sorts and views multiple lists of results to the user. The system can also inform the user when there are no matches of the searched journal. Every user can search as many as they require for journals of their interest.

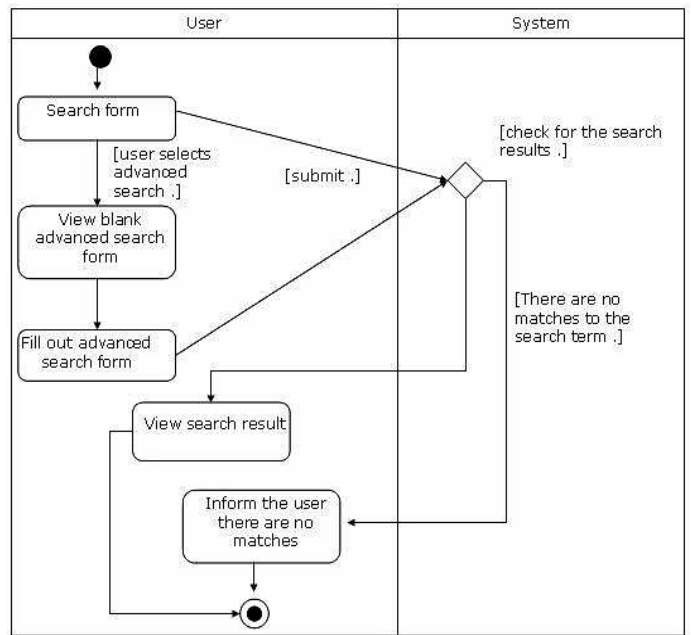


Figure 4: The Search Process Activity Diagram

4. Implementation and Results

This section provides the details of implementation followed by the discussion of the results.

4.1. Implementation

The WEKA Explorer version 3.9.4 was used for the training of data and implementation. This was done based on the data generated from the respondents' preferences discussed in section three. Naïve Bayes and Random Forest algorithms were used for the classification of the data and training of the model. The model generated was used as a knowledge-base for the development of the web-based journal recommender application. The data was saved in comma-separated values (CSV) format as required in the WEKA explorer and preprocessed by converting it into the .arff format, which is the format recognised by WEKA for a classification of this type. The preprocessed data were trained using the Naïve Bayes and Random Forests without any splitting. In classifying the data, accuracy was based on predicting the journal preferences class labels in the training data. Naive Bayes classifier is a major classifier that is very useful in Collaborative Filtering based Recommendation Systems. Naïve Bayes uses machine learning and data mining techniques to filter the information and use it to predict future occurrences. Naive Bayes algorithm can handle both continuous and discrete data. It requires less training data, can make probabilistic predictions, requires less training data, easy to implement and fast. Random Forest algorithm can be used for both classifications and regression task. It provides higher accuracy than other classification algorithms. Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data. If there are more trees, it will not allow overfitting trees in the model. Random Forests was chosen over decision trees and other classification methods because it is easy to interpret and make for straightforward visualizations, the internal workings are capable of being observed and thus make it possible to reproduce work, it can

handle both numerical and categorical data, and performs well on large datasets and are extremely fast.

4.2. Results and Discussion

Results of data training and classification carried out in WEKA explorer using the two ranking algorithms is presented in this section.

4.2.1. Results from Classification with Naïve Bayes Algorithm

Figure 5 shows the results of data training and model generation and with Naïve Bayes algorithm in WEKA explorer. Eight hundred and sixty-seven instances of the data and five attributes were trained using the 'Use training set' option.

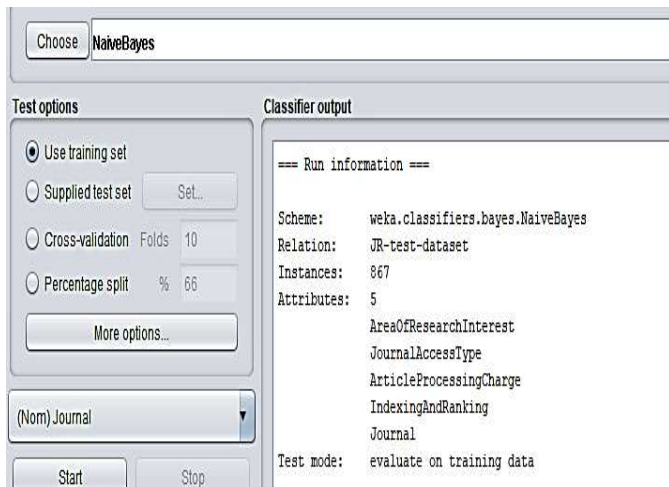


Figure 5: Details of Classification with Naïve Bayes Algorithm

Results from the classification using Naïve Bayes algorithm are shown in Figure 6. Time taken to build the model was 0.01 seconds. Time taken to test the model on training data was 0.02 seconds. There were six hundred and thirty (72.66%) correctly classified instances and two hundred and thirty-seven (27.34%) incorrect classified instances. Mean Absolute Error (MAE), which is a measure of the average magnitude of the errors in a dataset without considering their direction, was 0.0225. This result implied that the error associated with this classification was minimal and insignificant. Root Mean Squared Error (RMSE), which is a measure of the average magnitude of the error, was 0.0943. In this study, RMSE was larger than MAE, which implied that there was variance in the individual errors of the dataset. This is consonance with the standard as RMSE is always larger than or equal to the MAE. Kappa statistic, which measures the agreement of prediction with the actual class was 0.714. This showed a higher agreement between the predicted and the actual class. The evaluation measures that also showed the accuracy of the model are given as True Positive (TP) Rate, False Positive (FP) Rate, Precision, Recall, F-Measure, MCC, ROC Area and PRC Area. TP signifies the correct predictions. From the results obtained using Naïve Bayes classification, TP was an average of 0.727, which implied that more of the class labels were predicted correctly. False Positive (FP), which is the number of instances predicted positive that was negative, was an average of 0.014, which implied that fewer instances were not predicted correctly. Precision was an average of 0.789, which meant that many cases that were predicted

positive are positive. The recall was an average of 0.727, which indicated that more of the positive instances were predicted positive. F-measure was an average of 0.760, which indicated that the precision and recall were evenly weighted. MCC was an average of 0.765. ROC Area was an average of 0.977. PRC Area was an average of 0.785. The high values obtained for MCC, ROC Area and PRC indicated that the model was predicting with a very high level of accuracy.

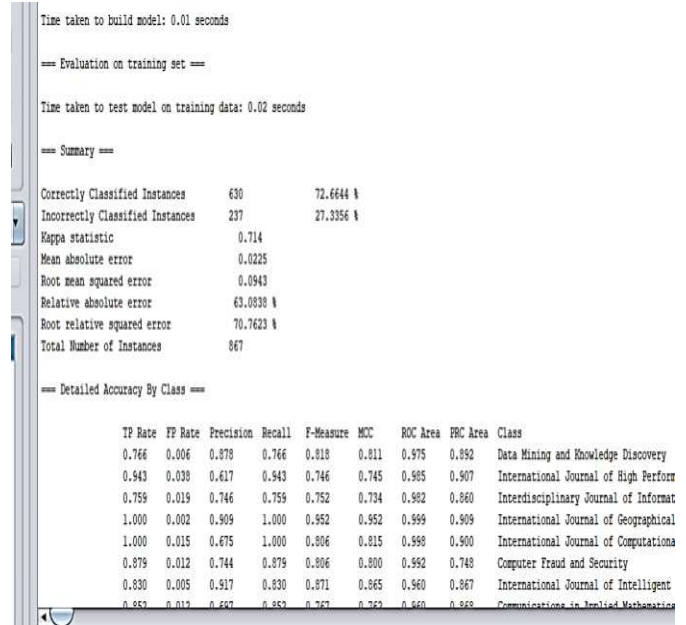


Figure 6: Results from Classification with Naïve Bayes Algorithm

4.2.2. Results from Classification with Random Forest Algorithm

Figure 7 shows the results of data training and model generation and with Naïve Bayes algorithm in WEKA explorer. Eight hundred and sixty-seven instances of the data and five attributes were trained using the 'Use training set' option.

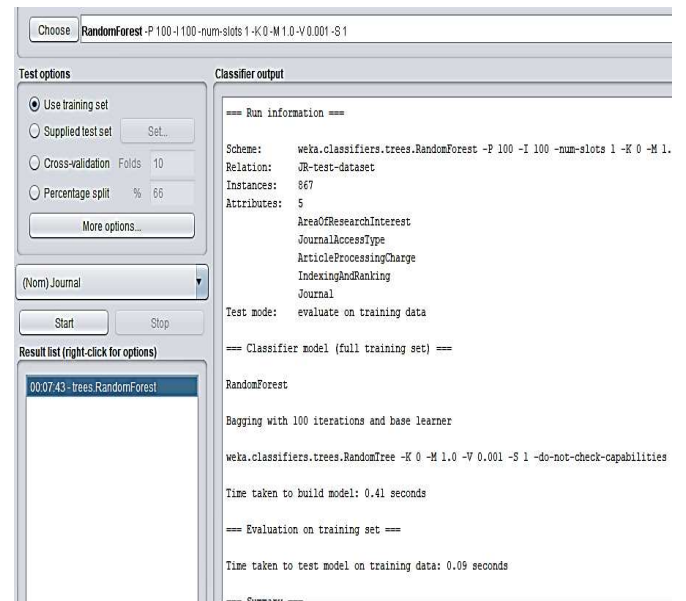


Figure 7: Details of Classification with Random Forest Algorithm

Results from the classification are shown in Figure 8. Time taken to build the model was 0.41 seconds. Time taken to test the model on training data was 0.09 seconds. There were seven hundred and seventy-eight (89.73%) correctly classified instances and eighty-nine (10.27%) incorrectly classified instances. Kappa statistic was 0.893, which represents a higher agreement between the actual and the predicted class. MAE was 0.0061 and RMSE was 0.0526. which implied that there was variance in the individual errors of the dataset. Other evaluation measures that also showed the accuracy of the model are TP, which was an average of 0.897, which implied that more of the class labels were predicted correctly. Fewer instances were incorrectly predicted with FP as 0.004. The recall was an average of 0.897 and precision was an average of 0.900, which meant that many cases that were predicted positive are positive. F-measure was an average of 0.886, which indicated that the precision and recall were evenly weighted. ROC Area was an average of 0.998, PRC Area was an average of 0.912 and MCC was an average of 0.892. The high values obtained for ROC Area, PRC and MCC showed that the prediction accuracy of the model was very high.

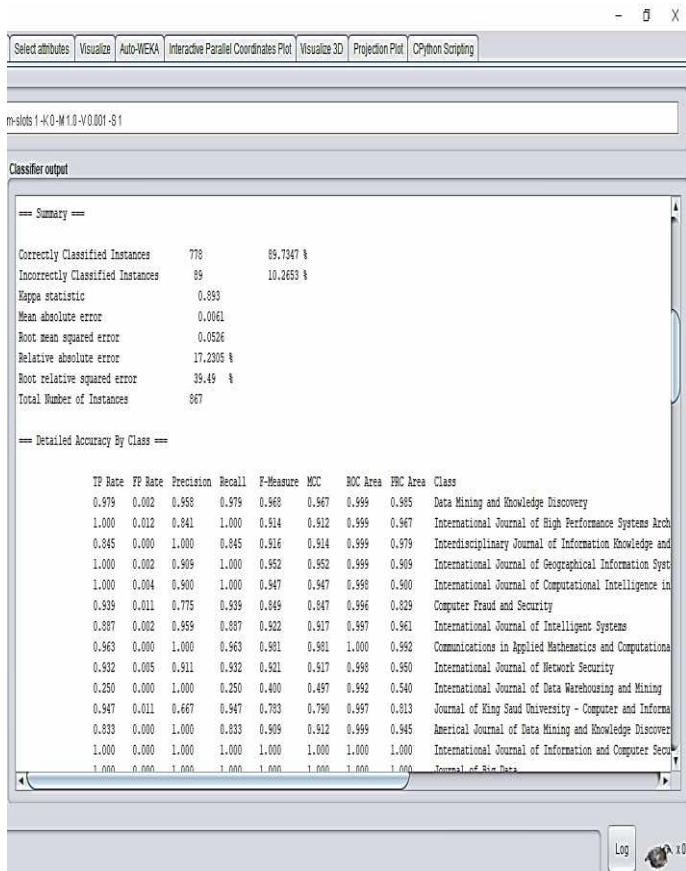


Figure 8: Results from Classification with Random Forest Algorithm

#### 4.2.3. Comparing Results of Classification with Naïve Bayes and Random Forest Algorithms

The performance metrics for comparing the data training and model generation of the Naïve Bayes and Random Forest algorithms are shown in Table 4. The Naïve Bayes classification was faster than Random Forest in model building and testing. On the other hand, the Random Forest algorithm outperformed Naïve

Bayes in terms of classification accuracy, Kappa statistics, true positives, ROC Area and all other performance metrics measured in this work.

Table 4: Results of Classification Using Naïve Bayes and Random Forest Algorithms Compared

PERFORMANCE METRICS	NAÏVE BAYES	RANDOM FOREST
Training time (s)	0.01	0.41
Testing time (s)	0.02	0.09
Instances correctly classified (%)	72.66	89.73
True positive	0.727	0.897
False positive	0.014	0.004
MAE	0.0225	0.0061
RMSE	0.0943	0.0526
Kappa Statistics	0.714	0.893
ROC Area	0.977	0.998
PRC Area	0.785	0.912
F-measure	0.760	0.886
Avg_Precision	0.789	0.900
MCC	0.765	0.892

#### 4.2.4. Evaluation of Results

In this section, the results obtained from the study were compared with existing methods, which were already discussed in section two. After comparing the classification accuracies, the classification accuracy derived in this work (89.73%) outperformed four related works [13], [16]-[18].

Table 5: Results of Classification Using Naïve Bayes and Random Forest Algorithms Compared

Reference	Year of Publication	Algorithm Used	Classification Accuracy (%)
13	2019	C4.5	78.84
16	2017	Time sequence-based algorithm	60
17	2018	CNN	75
18	2015	Neural Network	63
This work	2020	Random Forest	89.73

#### 4.3. System Interface

The section describes the features of the developed recommender system. The most important pages are described here and in the following subsections. The system's interface was built with HTML, CSS, PHP, Java scripts and AppServe. The interface for the application was thoroughly tested to ensure that it complies with the requirement of the recommendation system. The home page (Figure 9) is the first page a user is presented once the application loads. It shows the labels Home, Categories, Journals, Publishers, Recommendations, Register, Login and Search icons.

For example, users will click the 'Journals' menu to access the list of journals in the system and likewise click 'Recommendations' menu to make necessary recommendations after answering a few questions by selecting their options. The

'categories' menu shows the various fields of computer science available for searching.

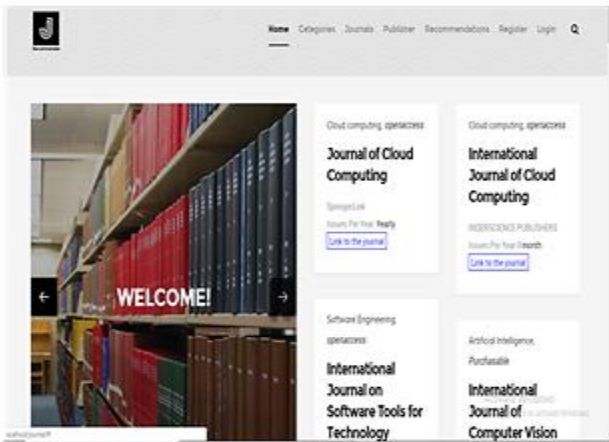


Figure 9: The home page of the 'Journal Recommender'

#### 4.3.1. The Registration Page

The 'Registration page' is the page where a user creates an account. In this page the system requests for Name, Phone number, Email address, the type of user, Research topic, Selection of the field of study, the type of the Journal access he/she is interested in patronising, that is, open access or purchasable. The Registration Page (Figure 10) is significant because recommendations are generated from preferences any user inputs into the system, and the information supplied here helps to build the system's database with more robust information useful for future suggestions or recommendations.

Figure 10: The Registration Page

#### 4.3.2. The Journal Page

The 'Journal page' (Figure 11) is a direct link to all the journals available in the recommender system, and it gets updated from time to time.

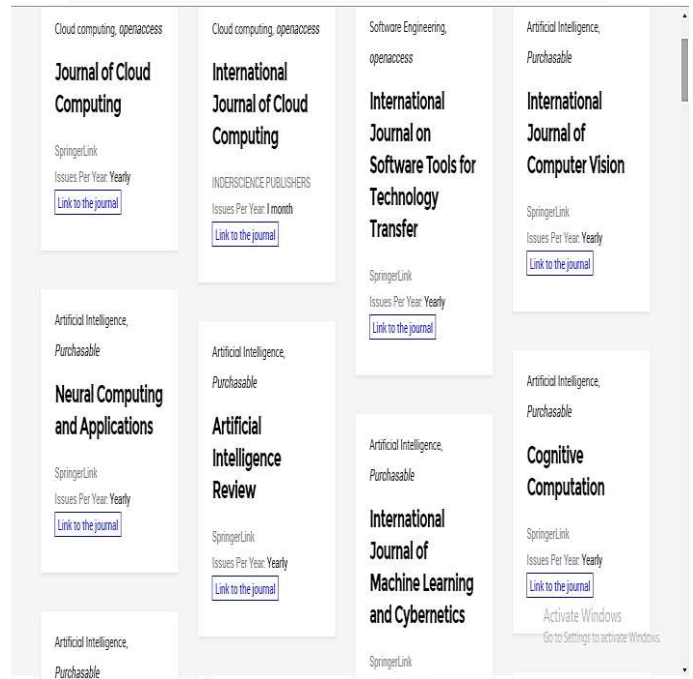


Figure 11: The Journal Page

#### 4.3.3. The Publisher's Page

This 'publisher's page' (Figure 12) consists of the complete list of journal publishers and the journals each of them contain in the developed system.

Name	Action
INDERSCIENCE PUBLISHERS	see journals
SpringerLink	see journals
Science Publishing Group	see journals
Scientific & Academic Publishing	see journals
SAGE Publishing	see journals
World Scientific	see journals
Microsoft Academic	see journals

Figure 12: The Publisher's Page

#### 4.3.4. The 'Get Journals Recommendation' Page

The 'Get Journals Recommendation' Page (Figure 13) is the main page displayed to the user when the "Recommendations" link is clicked. It requires the user to fill in the recommendation form, which is based on user's interests and selection of options. Some of the options that determine the final recommendation are the type of Journal Access, the kind of journals that serves user's interest,

the research topic, the amount of article processing fee that is charged and a few others. The user is expected to fill the form.

Figure 13: The Get Journals Recommendation Page

In Figure 13, the field “What Kind of Journals are you interested in?” refers to the type of journal access the user wants. Some journals are accessible to view, that is, open access, and others have to be purchased before viewing. The user is given a choice to choose the type of journal access he/she is interested in. Also, the field, ‘Article Processing Fee’, is also known as publication fee, which is charged to authors to make their work available in either the open access but free or restricted access and paid versions. This information is usually included in the Authors Instruction page of most journals. The range used for the purpose of the recommendation carried out in this work is from free, one to one-hundred USD, one hundred and one USD, two hundred and one to three hundred USD, three hundred and one to four hundred USD, four hundred and one to five hundred USD and five hundred USD & above. The category field defines what field of computer science the user is interested in. Examples of the fields in the drop-down list are Software Engineering, Information systems and Artificial Intelligence, just to mention a few. The rating field is used the system’s feedback mechanism based on the user’s acceptance of the recommendations. It has just two options, which are “Very Helpful” if the user is satisfied and “Not Helpful” if the user is dissatisfied with the recommendations from the system. This information is handy as the system makes use of it to self-improve itself from time to time.

#### 4.3.5. System’s extendability

The System Admin.’s page allows the flexibility and extendability of the system. The admin can add and update more publishers, more journals, more publishers, new users, review article processing charges and several other functions that extend the system’s robustness and effectiveness.

## 5. Conclusion

An academic researcher can be an author who wants his or her article, papers or life’s work to be published in journals through publishing groups. Over the years, the amount of information on

the internet is enormous, and this causes information overload problem to researchers during their publication quest to finding the right journals to publish their works. This study proposed a solution that applied a monolithic hybrid recommender system with feature combination and augmentation along with two algorithms, which are Naïve Bayes and Random Forest. Data used in this work were computer science-related journals, collected from pre-existing data that are readily available on academic web sites and repositories. Five attributes were used as preferences for the recommendation of journals. They are the fields of study, type of journal access, research topic, article processing fee and rating of the journals. These attributes were trained and modelled in a machine learning tool, WEKA, and the model generated was used to develop a front-end interface application where prospective academic researchers and authors can enter preferences and view journal recommendations directly generated by the developed system. This system was tested and was found to make recommendations with high levels of accuracy. The recommender system, developed in this work, will help researchers and authors to quickly and dynamically determine which journal is best suitable for their ready-to-be published articles. It is highly recommended that universities, researchers, authors could start adopting the system to improve publishing speed and output. Future works will involve the use of ensemble machine learning algorithms. A comparative analysis would also be done to test the best method under different scenarios. Finally, future works would also include recommendations of other significant disciplines aside from computer science.

## Conflict of Interest

The authors declare no conflict of interest.

## Acknowledgment

The authors appreciate the Department of Computer Science’s laboratory, Redeemer’s University, Ede, Osun State, Nigeria for providing a convenient environment for the analysis.

## References

- [1] J.A. Konstan, J. Riedl, “Recommender systems: From algorithms to user experience,” *User Modeling and User-Adapted Interaction*, **22**(1–2), 101–123, 2012, doi:10.1007/s11257-011-9112-x.
- [2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Systems*, **46**, 109–132, 2013, doi:10.1016/j.knosys.2013.03.012.
- [3] R. P. V. H.R., “Recommender Systems,” *Communications of the ACM*, **40**(3), 77–78, 1997.
- [4] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modelling and User-Adapted Interaction*, **12**(4), 331–370, 2002, doi:10.1023/A:1021240730564.
- [5] M.. Pazzani, Billsus D., “Content-Based Recommendation Systems,” *The Adaptive Web*, **43**(2), 325–341, 2007, doi:https://doi.org/10.1007/978-3540-72079-910.
- [6] U. Shardanand, P. Maes, “Social information filtering: Algorithms for automating word of mouth,” in *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, 210–217, 1995.
- [7] G. Adomavicius, A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, **17**(6), 734–749, 2005, doi:10.1109/TKDE.2005.99.
- [8] K. Madadipouya, Sivananthan Chelliah, “A Literature Review on Recommender Systems Algorithms, Techniques and Evaluations,” *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, **8**(2), 109, 2017.

- [9] C.C. Aggarwal, *Recommender Systems: The Textbook*, Springer International Publishing, Switzerland, 2016.
- [10] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, *Recommender Systems*, Cambridge University Press, Cambridge, 2010, doi:10.1017/CBO9780511763113.
- [11] I. Kubinyi, *Designing an Automatic Recommendation System for a Web Based Social e-Learning Application*, Helsinki Metropolia University of Applied Sciences, 2013.
- [12] A.O. Ogunde, E. Ajibade, "Adewale Opeoluwa Ogunde, Emmanuel Ajibade. A K-Nearest Neighbour Algorithm-Based Recommender System for the Dynamic Selection of Elective Undergraduate Courses," *International Journal of Data Science and Analysis*, **5**(6), 128–135, 2019, doi:10.11648/j.ijdsa.20190506.14.
- [13] A.O. Ogunde, J.O. Idialu, "A recommender system for selecting potential industrial training organizations," *Engineering Reports*, **1**(3), 2019, doi:10.1002/eng2.12046.
- [14] R. Ghani, A.E. Fano, *Building a Recommender System using a Knowledge Base of Product Semantics*, 2000.
- [15] W.W. Zhang, F.G. Liu, L. Jiang, D. Zu, "Recommendation Based on Collaborative Filtering by Convolution Deep Learning Model Based on Label Weight Nearest Neighbor," in *10th International Symposium on Computational Intelligence and Design*, 504–507, 2017.
- [16] S. Gore, J. Sonawane, L. Duggineni, "A Product Recommendation System Based on Hybrid Approach," **3**(5), 1–5, 2017.
- [17] A. Kumar, T. Sairam, "Machine Learning Approach for User Accounts Identification with Unwanted Information and data," *International Journal of Machine Learning and Networked Collaborative Engineering*, **2**(3), 119–127, 2018, doi:10.30991/ijmlnce.2018v02i03.004.
- [18] Y. Cai, R. Lin, Y. Kang, *A Personalized Company Recommender System for Job Seekers*, 2015.